Лабораторная работа №4

Продвинутое использование git.

Митрофанов Тимур Александрович

Содержание

# 1 Цель работы

Получить навыки правильной работы с репозиториями git.

# 2 Выполнение лабораторной работы

Разрешаем доступ к репозиторию в которм находится gitflow ***dnf copr enable elegos/gitflow***. (рис. 1)

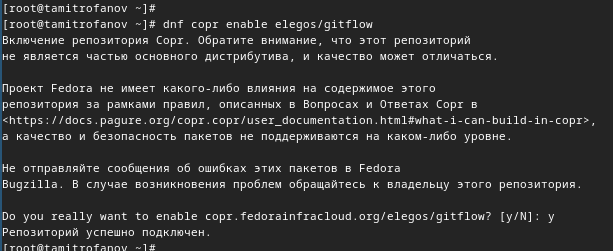


Рис. 1: Активация доступа к репозиторию

Устанавливаем gitflow, используя команду \***dnf install gitflow**. (рис. 2)

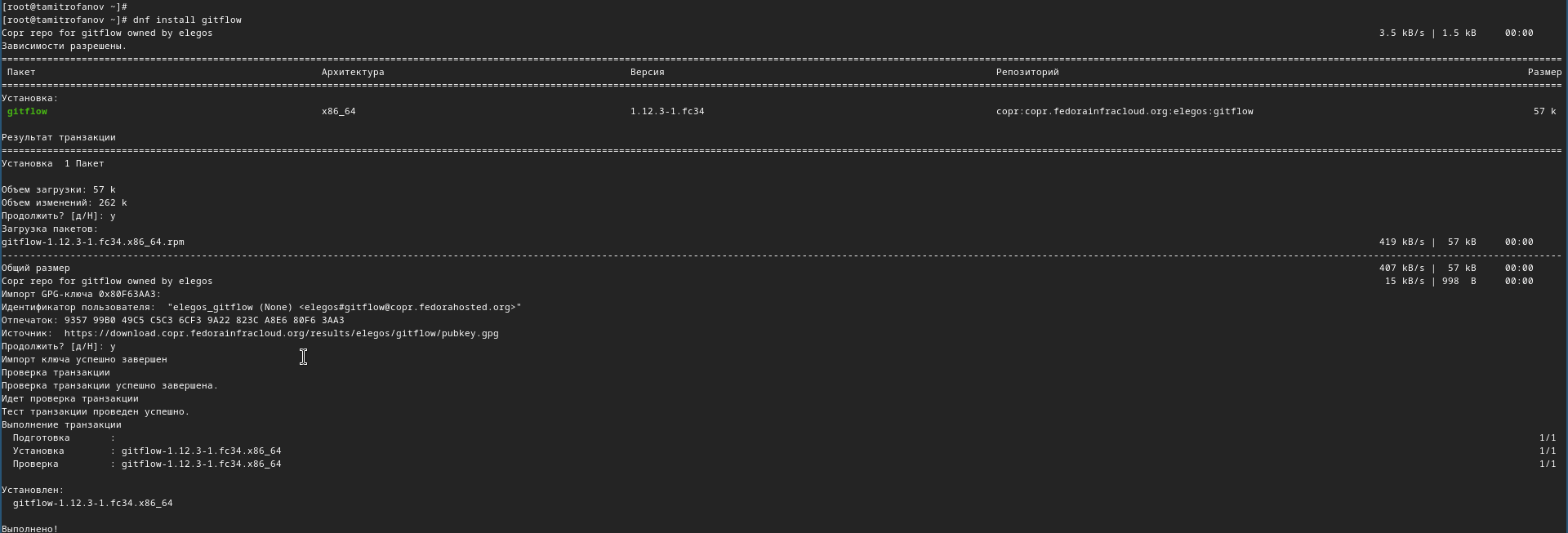


Рис. 2: Установка gitflow

Устанавливем Node.js и pnpm (рис. 3) и (рис. 4)



Рис. 3: Установка Node.js

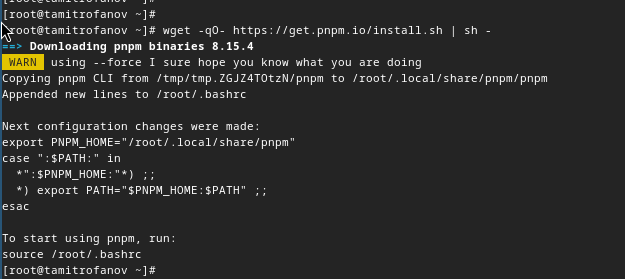


Рис. 4: Установка pnpm

Настраиваем Node.js для работы. (рис. 5)

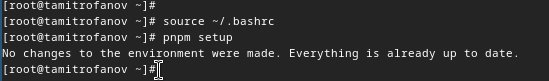


Рис. 5: Настройка Node.js

При помощи команды ***pnpm add -g commitizen*** устанавливаем программу для помощи в форматировании комитов. (рис. 6)

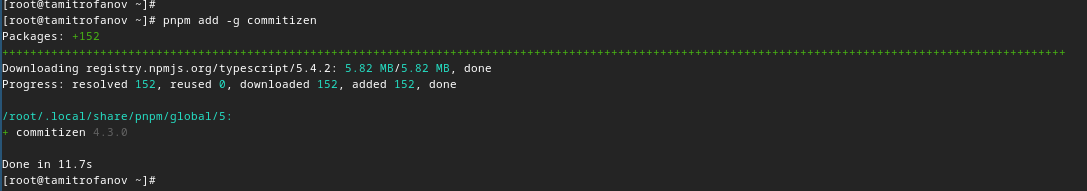


Рис. 6: Установка commitizen

При помощи команды ***pnpm add -g standard-changelog*** устанавливаем программу для помощи в создании логов. (рис. 7)

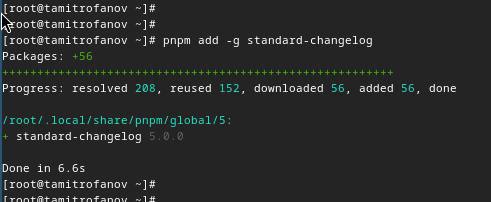


Рис. 7: Установка standard-changelog

Создаём репозитрий git-extended. (рис. 8)

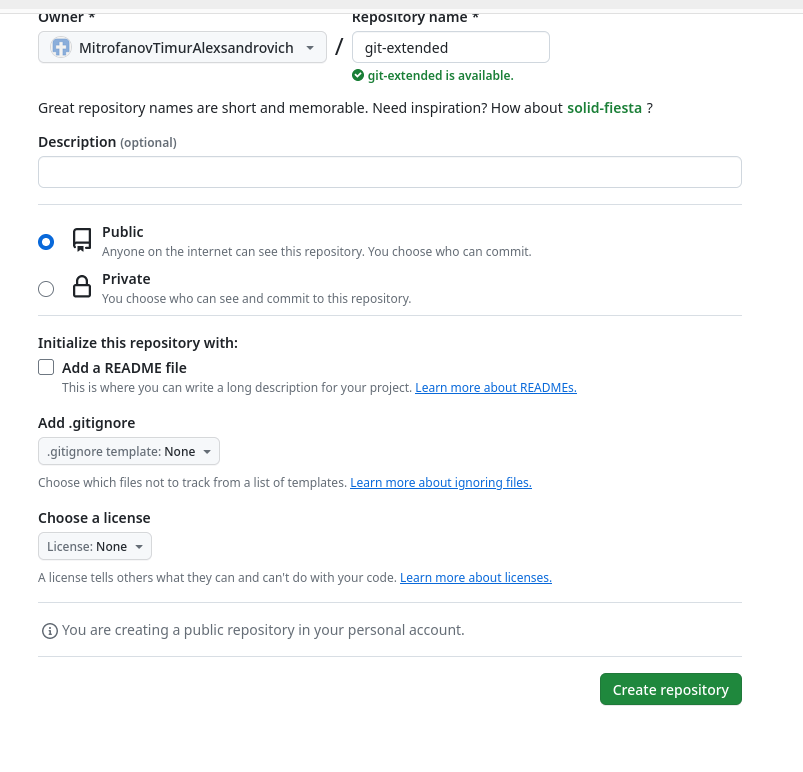


Рис. 8: Создание репозитория git-extended

Копируем внось созданный репозиторий себе на вирт. машину. (рис. 9)



Рис. 9: Клонирование репозитория git-extended

Делаем первый комит на репозиторий git-extended. (рис. 10)

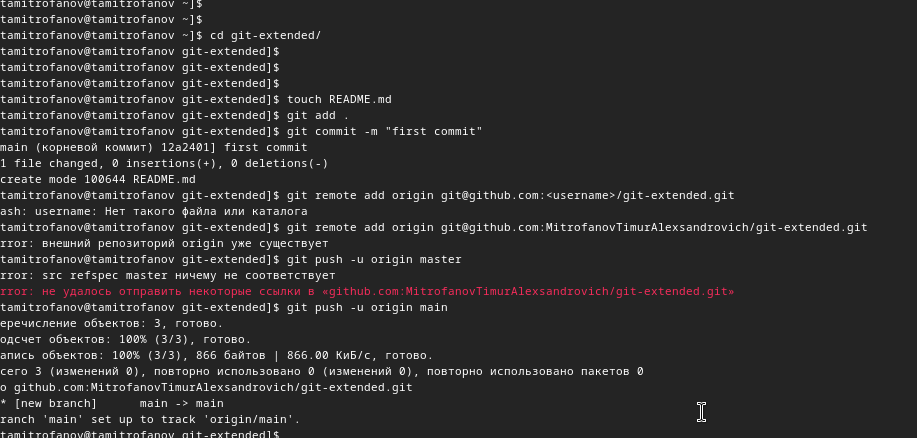


Рис. 10: Создание нового файла и сохранение его на сервер

Смотри конфигурацию пакетов Node.js. (рис. 11)

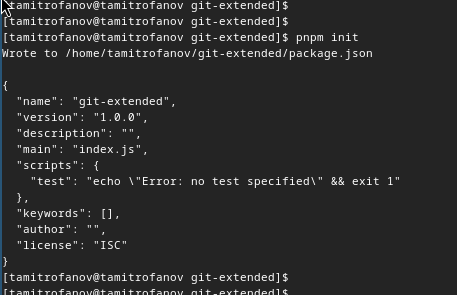


Рис. 11: Конфегурация пакетов Node.js

Модификация конфигурации Node.js. (рис. 12)

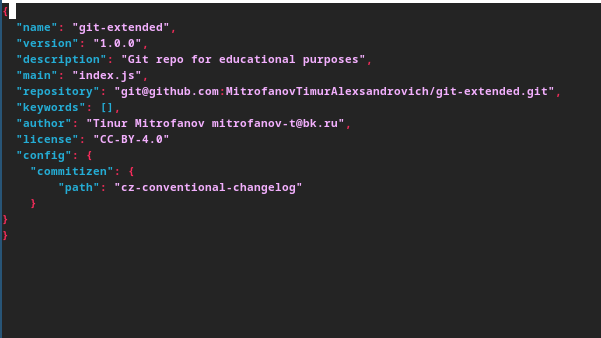


Рис. 12: Модификация конфигурации Node.js

Добавляем новые фалый, делаем коммит и отправляем на github. (рис. 13) и (рис. 14)

Добавляем файлы

Рис. 13: Добавляем файлы

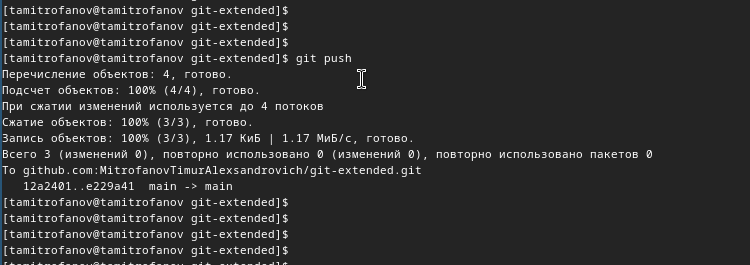


Рис. 14: Отправляем файлы на github

Инициализируем gitflow. (рис. 15)

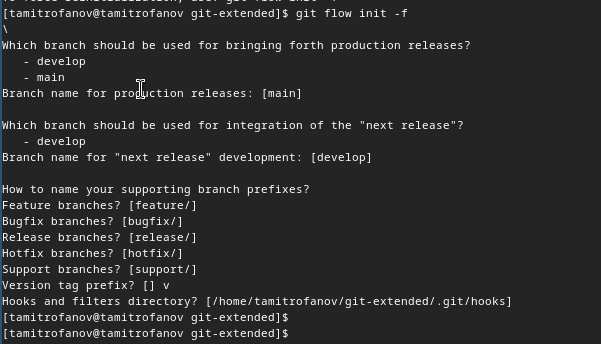


Рис. 15: Инициализация gitflow

Проверяем в какой мы ветке. (рис. 16)

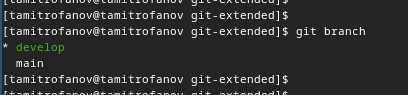


Рис. 16: Проверка выбранной ветки

Загружаем всего репозитория в хранилище. (рис. 17)

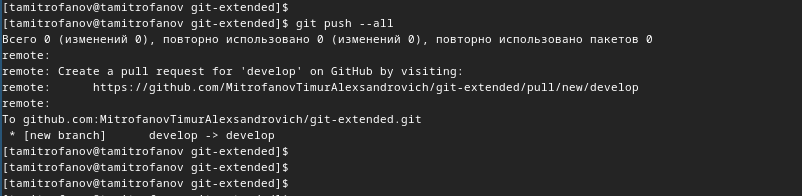


Рис. 17: Загрузка всего репозитория в хранилище

Установливаем внешнюю ветку как вышестоящую для этой ветки при помощи команды ***git branch –set-upstream-to=origin/develop develop***. (рис. 18)

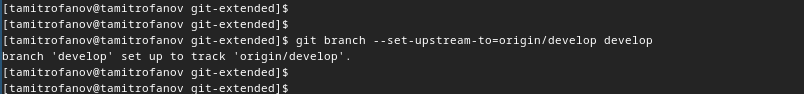


Рис. 18: Установка внешней ветки как вышестоящей для этой ветки

Создадим релиз с версией 1.0.0 при помощи команды ***git flow release start 1.0.0***. (рис. 19)

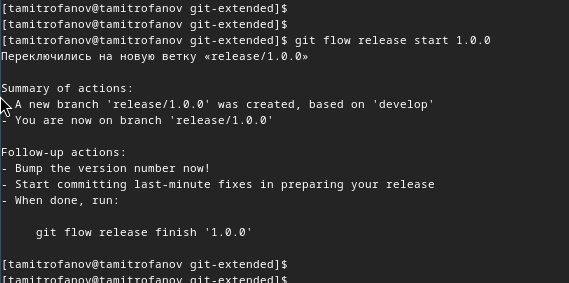


Рис. 19: Создание релиз версии 1.0.0

Создадим журнал изменений при помощи команды ***standard-changelog –first-release***. (рис. 20)

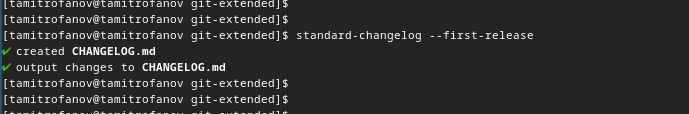


Рис. 20: Создание журнала изменений

Добавим журнал изменений в индекс при помощи команд ***git add CHANGELOG.md*** и ***git commit -am ‘chore(site): add changelog’***. (рис. 21)

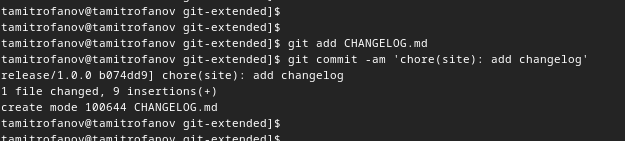


Рис. 21: Добавление журнала изменений в индекс

Зальём релизную ветку в основную ветку при помощи команды ***git flow release finish 1.0.0***. (рис. 22)

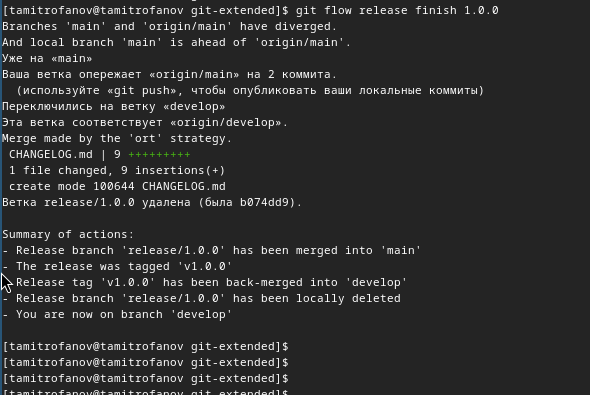


Рис. 22: Залив релизной ветки в основную ветку

Отправим данные на github. (рис. 23)

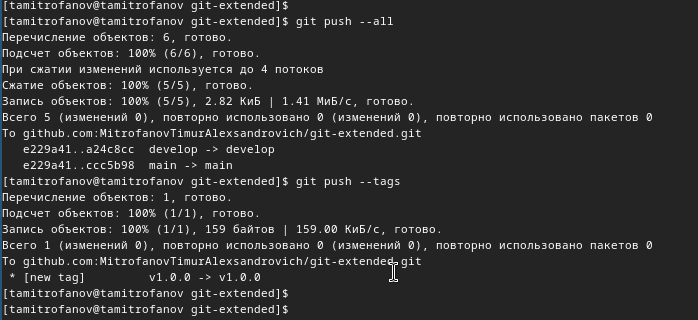


Рис. 23: Отправка данных на github

Создадим релиз на github. Для этого будем использовать утилиты работы с github. (рис. 24)



Рис. 24: Создание релиза на github

Создадим ветку для новой функциональности. (рис. 25)

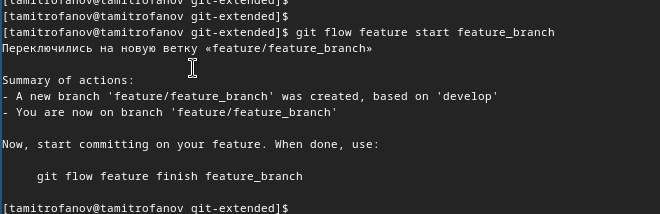


Рис. 25: Создание ветки для новой функциональности

По окончании разработки новой функциональности следующим шагом объединим ветку feature\_branch c develop. (рис. 26)

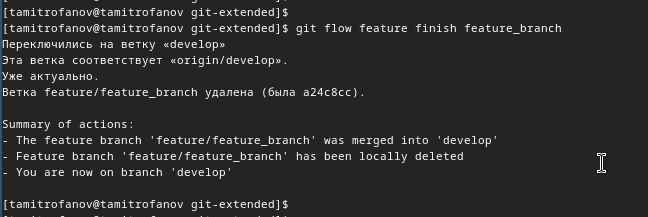


Рис. 26: объединение ветки feature\_branch c develop

Создадим релиз с версией 1.2.3. (рис. 27)

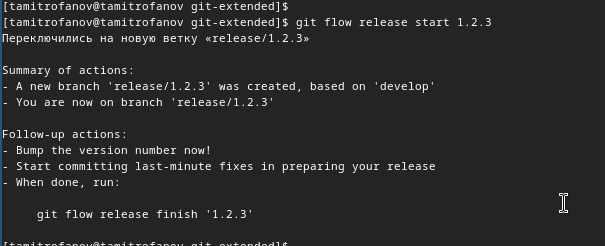


Рис. 27: Создание релиза с версией 1.2.3

Изменений файла конфигурации для соответствия версии релиза. (рис. 28)

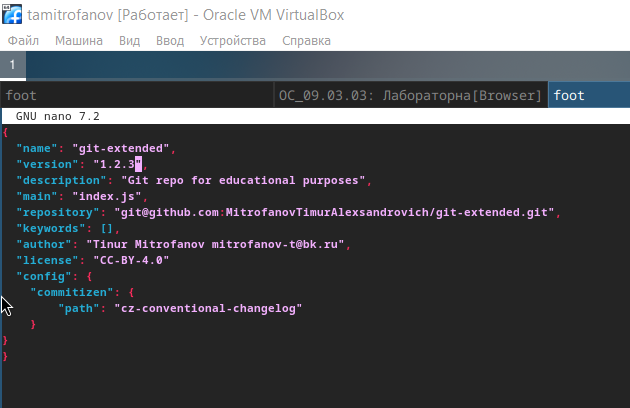


Рис. 28: Изменений файла конфигурации

Создадим журнал изменений. (рис. 29)

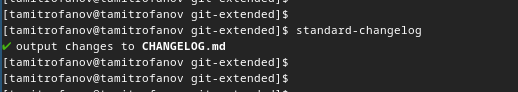


Рис. 29: Создание журнала изменений

Добавим журнал изменений в индекс при помощи команд ***git add CHANGELOG.md*** и ***git commit -am ‘chore(site): update changelog’***. (рис. 30)

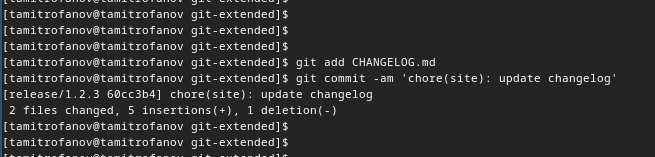


Рис. 30: Добавление журнала изменений в индекс

Зальём релизную ветку в основную ветку при помощи команды ***git flow release finish 1.2.3***. (рис. 31)

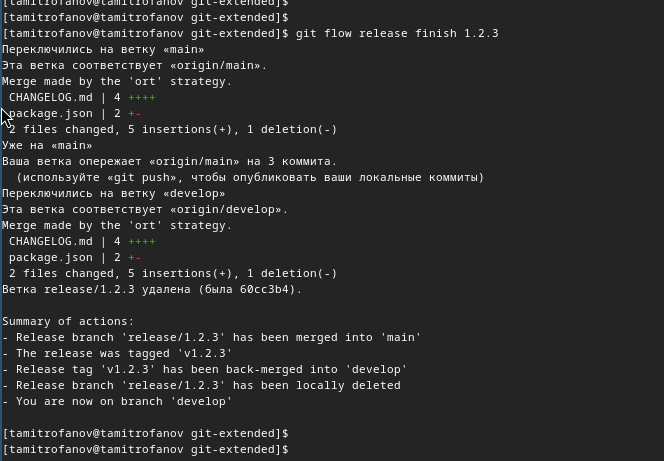


Рис. 31: Залив релизной ветки в основную ветку

Отправим данные на github. (рис. 32)

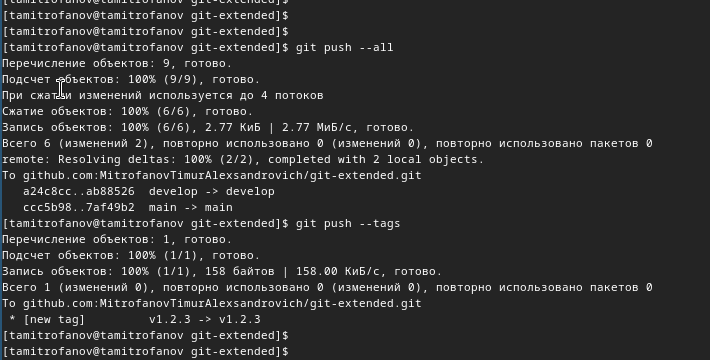


Рис. 32: Отправка данных на github

Создадим релиз на github. Для этого будем использовать утилиты работы с github. (рис. 33)

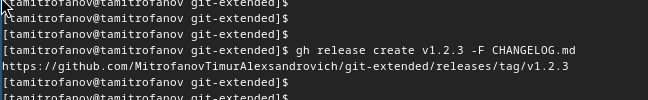


Рис. 33: Создание релиза на github

# 3 Выводы

В этой лабораторной работе я получить навыки правильной работы с репозиториями git.